Error-Propagation Analysis and Concealment Strategy for MPEG-4 Video Bitstream with Data Partitioning

Yung-Chi Chang, Chao-Chih Huang, Hao-Chieh Chang, Hung-Chi Fang, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

ABSTRACT

Data partitioning, one of the MPEG-4 video error resilience tools, enables better error robustness. However, it suffers from serious error propagation problem. In this paper, we use an experimental approach to model the error propagation with several commonly used error detection conditions. It is shown that errors detected in forward section of texture data may be propagated from motion data, while those in DCT coefficients mostly result from themselves. Furthermore, the motion marker is the major error source for several error conditions detected in motion part. According to these characteristics, motion marker assumption and backtracking-based concealment strategies are proposed to achieve more accurate error localization in bitstream domain.

1. INTRODUCTION

Universal accessibility is an important functionality of MPEG-4 [1], which targets at audio-visual coding in multimedia applications with interactivity, high compression, and portability of audio and video content. Audio-visual information should be accessible from anywhere in any way. This results from rapid growing variety of network, including mobile ones and Internet. However, bandwidth limitation and channel error characteristics can probably corrupt data transmitted over network. Therefore, robust transmission of compressed data over unreliable channels has become an important requirement. To achieve this goal, the decoding terminal should be able to perform error detection and error concealment furthermore. Several techniques [2][3][4] have been proposed for these capabilities when no error resilience tools are provided in encoding terminal. To accomplish error detection, characteristics of natural video signals are exploited in addition to bitstream syntax checking. For error concealment, both spatial and temporal domain data can be applied to improve quality.

For compressed multimedia data, data dependency becomes stronger because of redundancy reduction when compression ratio is higher. This increases the probability of data corruption and difficulties for data recovery. In order to provide better error robustness with minimal impact on coding efficiency, error resilience tools are used to add redundancy to bitstream. MPEG-4 Video standard [1] includes a lot of error resilience tools, such as video packet resynchronization, data partitioning, reversible variable length code (RVLC), and header extension code [5][6].

This paper is organized as follows. In Section 2 bitstream structure with data partitioning enabled is presented. In Section 3 the analysis of error propagation is discussed. Finally we summarize our major findings and outline our method for implementation.

2. DATA PARTITIONING

The basic idea of data partitioning is to partition the bitstream within a video packet into several parts, and codewords with the same characteristics are in same partition. In the following discussion, we focus on frame-based P-type VOP (video object plane) in MPEG-4. Under such circumstance, all motion related syntactic elements are placed in motion partition, while all block DCT related syntactic elements are placed in texture part.

2.1 Bitstream Structure

Fig. 1 shows the bitstream structure within a P-VOP video packet. A data partitioned video packet is separated into packet header, motion data, and texture data. The packet header records MB number, quantizing scale and header extension code with repeated VOP information. The MB number field denotes the first MB number in this packet. It can be used to relocate the decoding after an error was found. The quantizing scale is a 2-bit fixed-length code used as an offset for the quantizer. The 1-bit header extension code indicates if there is a set of information following. The information includes VOP coding type and other important VOP parameters.

Following the video packet is motion data. It contains not_coded, MCBPC, and motion vectors of every MB. The not_coded bit indicates whether the MB is coded. If yes, the MB is decoded with zero motion vector and no DCT coefficients are included. If no, just after the bit is a variable-length MCBPC field to denote the MB type and chrominance-block coded pattern. Several MB types are for various profiles in MPEG-4 video. The number of motion vector depends on the MB type.

To separate motion data and texture data, a 17-bit motion marker is inserted between these two parts. The motion marker is a unique pattern different from any combination of MVD VLC codewords such that decoding terminal can distinguish it from other codes and begin texture part parsing.

Texture data includes block type information and DCT coefficients. The block type information includes AC prediction flag and CBPY. The AC prediction flag indicates the technique of AC prediction from former decoded MBs. CBPY is a variable-length code, recording the coded block pattern of the luminance components. The DCT coefficients are coded with reversible or non-reversible VLC. At the tail of the packet is the variable-length stuffing code. By the stuffing code, next packet starts at byte-aligned location, and the beginning of reverse decoding of DCT coefficients when RVLC is applied is guided.

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

IEEE COMPUTER SOCIETY

2.2 Previous Work

The concealment strategy when data partitioning is applied is described [5]. All MB data in current packet are discarded and replaced if an error is detected in motion part. When an error is detected in texture part, the data in motion part are regarded as correct. Compared with traditional bitstream structure within a video packet that organizes motion data and DCT data together in MB order, it is not necessary to always discard whole motion data when an error is detected. Besides, more stringent check can be performed on motion data by means of the location of the motion marker. Since errors in motion data affect video quality much more than texture data errors, it's expected that video quality under erroneous transmission channels can be enhanced. According to its simulation results, it improves the quality of compressed video data by over 2 dB.

However, the effect that error occurs in one or some fields in bitstreams may cause the following fields unrecognized by decoder is not discussed in these papers. This effect is called error propagation. This results in two serious problems. First, since the exact error corrupted location is unknown, the whole part where error is detected, which may contain correct information mostly, is discard. Besides, the error detected in texture part may propagate from motion part. But only texture data is discarded. This makes decoding terminal regard a lot of wrong motion data as correct, which causes video quality degradation. Hence, it is important to consider the error propagation issue in real case. To explore the characteristics of error propagation, experiment and analysis are performed and discussed in next section.

3. ERROR-PROPAGATION ANALYSIS

The error propagation analysis is performed with the following parameters:

- Visual profile: Simple
- Sequences: bream, children, foreman, news, weather
- Sequence size: CIF, 300 frames
- Packet length: 2048 bits
- Error type: random error (1 bit error in a video packet)

3.1 Analysis Approach

The approach of error propagation analysis is discussed in this section. At first, several error detection rules are defined to identify whether an error is detected. Each error detection rule defines its corresponding an error condition. The error conditions are listed in Table 1.

Table 1. Error conditions

Error Conditions	Description
0	Illegal MCBPC VLC codeword
1	Impossible MB type
2	Illegal MVD VLC codeword
3	Illegal CBPY VLC codeword
4	Illegal DC VLC codeword
5	Illegal DCT VLC codeword
6	More than 64 DCT coefficients
7	Invalid stuffing code

These conditions are classified into two categories: illegal VLC codeword, and unreasonable or impossible codeword data. EC0~EC2 are detected in motion part, while others are detected in texture part.

When any one of these error conditions is met, an error is detected. With these conditions, the experiment is performed with corrupting only one bit in the specified field randomly. In addition to the error condition caused by the error, the locations where error occurs and is detected are recorded. It is repeated for several times to obtain various possibilities for different fields. The pseudocode of the experiment flow is described below:

```
for (i = 0; i < total number of field in bitstream; i++)
{
   if (total bit number of field i in the bitstream > 1000)
      test count = 1000
   else
      test count = total bit number of field i in the bitstream

   for (j = 0; j < test count; j++)
   {
      one bit in the field is corrupted
      the corrupted location is recorded
      the erroroneous bitstream is parsed

      if (error is detected)
         detected location and error condition is recorded
   }
}
```

Figure 2. Pseudocode of the experiment flow

After the experiment is completed, the error source distribution for every error condition is calculated. The contribution of each field to a specific error condition depends on its total bit number in the bitstream and the percentage of detection in that error condition. Then, for a specific error condition, its error source distribution can be acquired by calculating the percentage of each field's contribution. The pseudocode of the calculation flow is described below:

```
count(i, k) = $EC_k$ count of error source field i

for (i = 0; i < total number of field in bitstream; i++)
{
   for (k = 0; k < total EC number; k++)
      weighted count(i, k) =
      bit number of field i in bitstream * count(i, k) / test count
}

for (k = 0; k < total EC number; k++)
{
   total weighted count(k) = $\Sigma_i$ weighted count(i, k)
   for (i = 0; i < total number of field in bitstream; i++)
      error contribution percentage(i, k) =
      weighted count(i, k) / total weighted count(k)
}
```

Figure 3. Pseudocode of the calculation flow

3.2 Analysis Results

From the experiment and calculation, the error source field (ESF) distribution for every error condition is acquired. At first, the distribution for the error conditions detected in motion part are shown in Table 2~4.

Table 2. Error source field distribution for EC0 (%)

ESF	bream	children	foreman	news	weather
Hori. MVD	24.33	29.14	22.40	9.61	17.01
Motion Marker	22.04	18.16	32.40	56.24	39.39
Not Coded	5.44	2.46	4.60	6.05	5.06
MCBPC	26.99	23.18	15.55	11.39	14.98
Vert. MVD	20.90	26.84	24.70	14.65	22.68
Others	0.30	0.21	0.34	2.06	0.87

Table 3. Error source field distribution for EC1 (%)

ESF	bream	children	foreman	news	weather
Hori. MVD	13.21	16.99	4.98	0.00	0.00
Motion Marker	57.15	56.30	76.73	84.78	81.70
Not Coded	2.83	2.94	3.01	0.00	0.00
MCBPC	10.32	12.98	8.62	2.56	11.00
Vert. MVD	15.69	9.66	4.62	0.00	2.37
Others	0.79	1.13	2.05	12.67	4.92

Table 4. Error source field distribution for EC2 (%)

ESF	bream	children	foreman	news	weather
Hori. MVD	26.05	24.75	24.33	1.16	9.99
Motion Marker	24.99	30.72	38.73	88.83	56.73
Not Coded	3.24	3.42	2.76	0.00	6.06
MCBPC	23.63	23.22	12.50	8.72	13.12
Vert. MVD	21.70	17.60	21.19	0.00	12.95
Others	0.39	0.28	0.49	1.28	1.14

From the above tables, it is obvious that the motion marker is the major ESF under these error conditions in most cases. For the error on motion markers, it's observed that most of them are detected within the marker from experimental results. The percentages are shown in Table 5.

Table 5. Self-detected percentage for motion marker (%)

SDP	bream	children	foreman	news	weather
EC0	96.84	97.80	76.64	91.26	91.41
EC1	70.79	74.60	47.37	68.33	57.69
EC2	96.08	94.92	80.89	83.72	87.16

Besides, the longest propagation length (LPL), which is the distance between corrupted location and detected location in bits, for ESFs in motion part under these three error conditions is shown in Table 6.

Table 6. Longest propagation length of error source field in motion part (bits)

LPL	bream	children	foreman	news	weather
EC0	201	187	181	72	156
EC1	70	137	95	5	31
EC2	197	182	136	27	104

Moreover, for the errors caused in fields in motion part, most of them are detected in texture part. That is, the error is propagated from motion part to texture part. The percentage of these errors detected in texture part among detectable ones from experimental results is shown in Table 7.

Table 7. Percentage for detected errors in texture part (%)

ESF	bream	children	foreman	news	weather
Not_Coded	89.41	93.35	92.38	99.26	97.47
MCBPC	77.69	70.44	88.19	95.18	90.89
Hori. MVD	72.96	54.27	87.64	95.00	88.11
Vert. MVD	73.44	57.62	87.22	93.02	84.65

For EC5 and EC6, it's observed that most of the ESF is DCT VLC codewords. The percentage is shown in Table 8.

Table 8. Percentage for error source field as DCT VLC codewords (%)

	bream	children	foreman	news	weather
EC5	94.75	96.95	94.80	79.78	88.71
EC6	81.28	89.90	76.11	72.97	78.36

Under EC5 and EC6, the LPL of DCT VLC codewords is listed in Table 9.

Table 9. Longest propagation length of DCT VLC codewords

LPL	bream	children	foreman	news	weather
EC5	495	231	280	372	206
EC6	425	455	534	528	350

4. CONCEALMENT STRATEGY

According to the analysis results, concealment strategies for these error conditions are summarized as follows:

- EC0, EC1, EC2

Since most of them are caused by errors in motion marker, which don't propagate very far, taking the detected location as motion marker's location is a reasonable solution. We should check backward for several bits to see if a code whose hamming distance with motion marker is 1 exists. If such a code exists and its location is the end the motion data of a MB, it is regarded as motion marker and bitstream parsing is continued. If it doesn't exist, we should backtrack for several bit length and discard the following data. The backtracking bit length can be acquired by referencing Table 6, but it shouldn't be longer than the distance between detection location and resynchronization marker of this packet. The percentage of the situation that the found motion marker is true motion marker is shown in Table 10. The backtracking length is 25 bits.

Table 10. Percentage of correctly found motion marker

	bream	children	foreman	news	weather
EC0	96.84	97.80	76.64	91.26	91.41
EC1	70.79	74.60	47.37	68.33	57.69
EC2	96.08	94.92	80.89	83.72	87.16

- EC5, EC6

These error conditions are DCT VLC codewords-related. Therefore, resulting from Table 10 and 11, backtracking for several bit length and discard the following data can be a better choice. The backtracking bit length can be acquired by referencing Table 11. It should be noted that remaining data that can't form a complete MB or block after backtracking should also be discarded. Compared with the approach that discards the entire texture part when error is detected, the average bit number and MB number saved by proposed strategy is listed in Table 11. The backtracking length is assumed to be 400 bits.

Table 11. Average saved bit number and MB number

	bream	children	foreman	news	weather
Bit Number	501.8	433.6	517.8	521.4	491.7
MB Number	6.9	7.6	7.6	50.1	16.3

5. CONCLUSION

Data partitioning is an error resilience tool adopted in MPEG-4 video standard. It provides better error robustness because it separates motion and texture data in a video packet. This paper addresses the error propagation problem under data partitioning mode. Bitstreams with random error are parsed and the error source field distribution for different conditions are experimented and calculated. The experimental results for various error conditions are presented and discussed. According to the results, some strategies for concealment in bitstream domain are proposed. The proposed strategy can achieve more accurate error localization with higher possibilities by motion marker assumption and backtracking under different error conditions.

6. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, *N2502a, Generic Coding of Audio-Visual Objects: Visual 14496-2, Final Draft IS*, Atlantic City, Dec. 1998.

[2] Wen-Jeng Chu and Jin-Jang Leou, "Detection and Concealment of Transmission Errors in H.261 Images," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 1, pp.74-84, February 1998.

[3] Han-Chiang Shyu and Jin-Jang Leou, "Detection and Concealment of Transmission Errors in MPEG-2 Images—A Genetic Algorithm Approach," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 6, pp.937-948, September 1999.

[4] Yao Wang, and Qin-Fan Zhu, "Error Control and Concealmet for Video Communication: A Review," *Proceedings of the IEEE*, Vol. 86, No. 5, pp.974-997, May 1998.

[5] Raj Talluri, Iole Moccagatta, Yashoda Nag, and Gene Cheung, "Error Concealment by Data Partitioning," *Signal Processing: Image Communication*, Vol. 14, Issue 6-8, pp. 505-518, May 1999.

[6] Luis Ducla-Soares, Fernando Pereira, "Error Resilience and Concealment Performance for MPEG-4 Frame-based Video Coding," *Signal Processing: Image Communication*, Vol. 14, Issue 6-8, pp.447-472, May 1999.

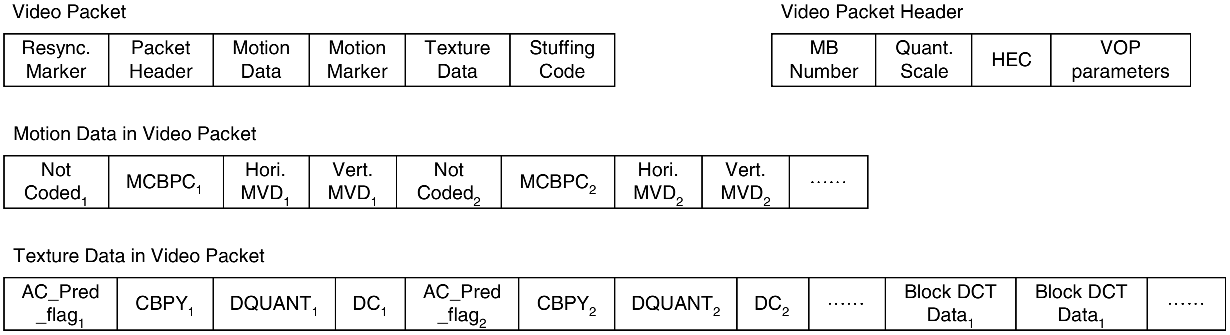

Figure 1. Bitstream structure in a P-VOP video packet with data partitioning

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE